S3,Taking Machine Intelligence to the Next, Much Higher Level
Version of November 25, 2010

Patrick Henry Winston

The Vision: Understanding Human Intelligence is the Great Enabler

We have increasing—and technically exploitable—understanding of why humans are smarter than other primates. That increased understanding of human intelligence includes an appreciation of the deep interaction between vision and language. On the vision side, humans have dedicated representations and processes for the recognition, analysis, and imagination of events. On the language side, humans have representations and processes that enable the description and re-description of events, that construct and exploit cases, and that make generalizations.

Going toward perception, *language enables the marshaling of visual and motor perceptions and actions*. Shout "watch out for the car on the right" and eyes go to the right, consequent to the listener's language faculty instructing the visual faculty. Demand "pick up the hammer" and the listener's hand goes out to follow the instruction.

Going toward thinking, *language enables description and description enables story telling and understanding*, and story telling and understanding lie at the center of human education. Story telling starts with our early exposure to fairy tales that keep us from wandering into the woods and goes on to our later reading of literature and history, and then to our still later personal experiences in life and surrogate experiences in law, medicine, business, and military schools.

Going toward both perception and thinking, *language enables imagination*, and the deployment of visual and motor perceptions and actions on situations never directly experienced. "You should never wear gloves when you use a table saw. Here is why: Your glove could get caught in the blade." Now, that is enough to stimulate your imagination. No further explanation is needed because you imagine what would follow. It does not feel like any sort of formal reasoning. It does not feel like you will have to have the message reinforced before it sinks in. It feels like you witness a grisly event of a sort it is likely no one has ever told you about. You have learned from a one-shot surrogate experience, and you are unlikely to wear gloves in the future when you operate a table saw.

Thus, language does much more than facilitate communication and enable the facile acquisition of syllogistic facts (if your heart stops, you die). If we are to understand how to make truly intelligent machines, we have to understand the role of language in story exploitation, in directing the perceptual and motor apparatus, and in the synthesis of situations that never occurred and the subsequent exploitation of those synthesized situations using all our

faculties from top to bottom, from story understanding to computation in the perceptual and motor faculties.

From this perspective, some of us who do Artificial Intelligence research focus too exclusively on symbolic reasoning. Others concentrate too much on symbol-free systems, such as neural nets and genetic algorithms. And still others find interest mainly in bulldozer computing, limiting themselves largely to statistical methods. All approaches contribute, but none addresses the central obstacles.

To take machine intelligence to the next level we need systems that understand sentences, so as to understand stories, and that put language in harness with our senses to understand the world, hence our S3 acronym.

Why now?

From a scientific perspective, the time is right because serious clues have emerged from biological research, because we have worked up important illustrations of concept, and because we can put into harness previously inconceivable computing resources.

The time is also right because other approaches to machine intelligence are just about played out, leaving a new generation of students eagerly looking to make contributions that are exciting and lasting. The best of the other approaches have contributed, but cannot go much further for lack of attention to the perception–language–story triumvirate.

That said, to take machine intelligence to the next level will require deep thinking and scientific innovation. Excessive early focus on demonstration problems will encourage engineering shortcuts and retard real progress.

What we leave out

Our emphasis on language and vision are two-thirds of a triad that ultimately must, we believe, involve contributions from the motor faculty, because the motor faculty grounds our understanding of how we volitionally affect the world.

Illustrations of concept

By way of illustration, we enumerate some enabling examples from MIT work, sponsored variously by NSF, AFOSR, DARPA, and ONR.

Biological Grounding

All the work in described in this section is heavily grounded in what is known about natural systems. For example, results from Elizabeth Spelke's work in developmental psychology and Ray Jackendoff's work in linguistic semantics led to the emphasis on language and guide the way in which we approach language. Similarly, results from Shimon Ullman's work on visual routines and Sajit Rao's work on visual attention, both inspired by many psychophysical studies, guide our vision work.

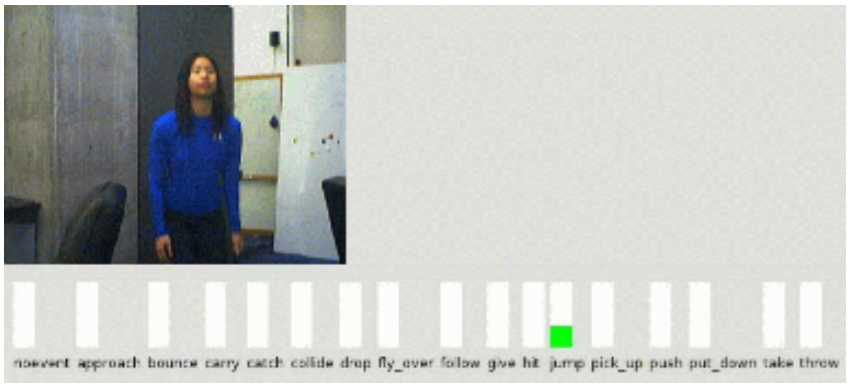

Figure 1. The Genesis/Vision system has learned to recognize *jump* and other actions from a few examples of each. The trained system recognizes that the student is jumping, as indicated by the green bar.

Genesis/Vision: A real-world problem solver

Taking the view that our human vision system is a problem solver, not just an input channel, it makes sense to transform a language-supplied problem into visual space if the problem is easier to solve over in that visual space. Thus, we draw pictures and imagine scenes for the same reason that leads an electrical engineer to perform a Fourier transform.

The Genesis/Vision side of our Genesis system sees the world through programs that recognize actions, and more recently, learn to recognize actions, such as approach, bounce, carry, catch, collide, drop, fly over, follow, give, hit, jump, pick up, push, put down, take, and throw. Figure 1 shows a student observed during a recognized jump.

Once, when Genesis/Vision seemed sluggish, a student jumped a second time, hoping to get a response. Genesis/Vision reported that the student was bouncing, having been trained on a bouncing ball. We were delighted by its generalization.

We believe that such grounding is essential to intelligence. A program limited to the symbolic manipulation cannot fully understand what it means to *give* when limited to the symbols alone because perception answers questions unanticipated by purely symbolic systems. Also, a program limited to symbolic manipulation cannot know as much because perception enables the acquisition of enormous amounts of world knowledge.

In summary, we believe that if we are to develop a computational theory of human intelligence, we must understand how to connect language to perception so that language and perception can work together to solve problems.

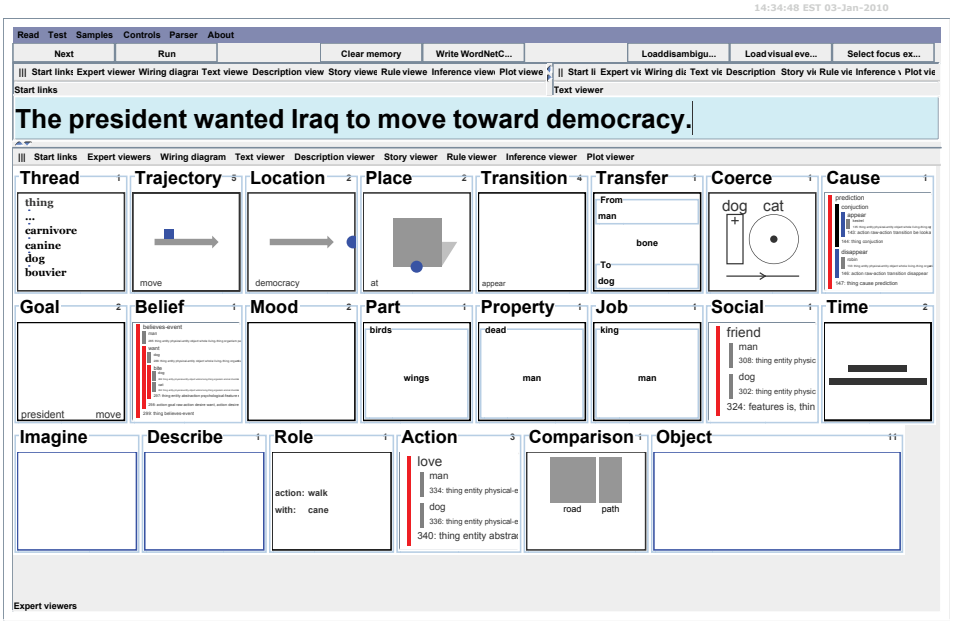

Figure 2. The Genesis/Language system features a large suite of representations, Threads, trajectories, and transitions appear frequently in written and spoken natural language.

Genesis/Language: A window on representation grounded in the physical world

On the sentence level, the Genesis/Language system sees the world through nearly two dozen frame-like representations.

As shown, a test suite of sentences instantiates, for example, representations for threads (approach to classification from Greenblatt and Vaina), trajectory (inspired by Jackendoff), transition (inspired by Borchardt) , transfer, location, time, cause, and coercion. There are many representations, in part, because there are many kinds of events to be described.

English descriptions instantiate these representations when we talk of physical-world events (the bird flew to a tree) as well as when we talk of abstract-world events (the country moved toward democracy).

The particular representations we use were gathered, in part, from work by linguists and researchers in Artificial Intelligence.

Others came from our own data-driven need to reflect the meanings encountered in the stories we use to drive our work.

Genesis work is representation-centric because we need representations to capture the constraints and regularities out of which we can build models, which in turn make it possible to understand, explain, predict, and control. Also, the bias toward multiple representations is inspired, in part, by Marvin Minsky's often articulated idea that if you have only one way of looking at a problem, you have no recourse if you get stuck.

The path from sentences to instantiated representations goes through the Start Parser, developed over a 25-year period by Boris Katz and his students. We have used other, statistically trained parsers, but Start has two compelling advantages: Start blunders less and Start produces a semantic net, rather than a parse tree, making it much easier to instantiate our frame-like representations. We also exploit WordNet, using it as a source of classification information. Of course, we could get by without WordNet by supplying classification

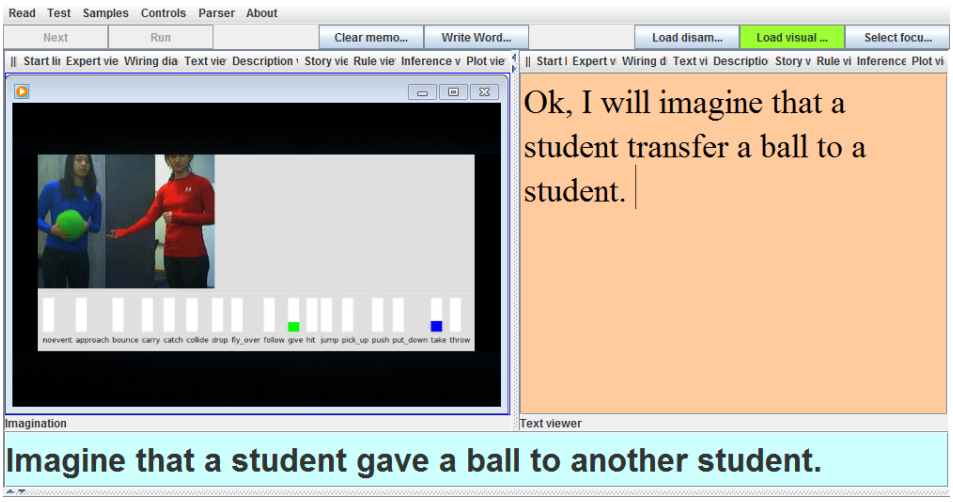

Figure 3. The Genesis/language system recalls a situation in which one student gives a ball to another. Because the Genesis/vision system sees a *take* in the same sequence, Genesis/language notes that *give* and *take* co-occur.

information in English (a Bouvier is a kind of dog) or by discovering it. Using WordNet is a temporary, time-saving shortcut.

In summary, we believe that if we are to develop a computational theory of human intelligence, we must focus on semantic analysis, treating language first as a means for describing what happens in the physical world, and then as a means for describing abstract worlds in which movement takes place in, for example, possession, emotion, power, and influence spaces.

Genesis/Language and Genesis/Vision: An essential partnership

We have demonstrated, although not regularly, that we can ask Genesis/Language to ask Genesis/Vision to answer a question by imagining an action. Suppose, for example, you say that a student gave a ball to another student, and then ask if the other student took the ball. A computer may solve the problem symbolically, knowing a rule such as: X took the ball because Y gave the ball to X.

Our Genesis/Vision system solves the problem with imagination, using visual routines that read the answer off of a stored, then-recalled scene. The vision side of Genesis/Vision recalls the following scene because, when analyzed visually, the *give* bar lights up. Then, it answers the *take* question by noting that the same scene lights up the *take* bar as well:

Our immediate aim is to make this kind of exchange and generalization regular and always on. We want Genesis/Vision to seek opportunities to learn by observing, and we want Genesis/Language both to serve Genesis/Vision by asking humans questions and to direct Genesis/Vision by telling it to attend to particular events.

In summary, we believe that if we are to develop a computational theory of human intelligence, we must arrange for perceptual and symbolic systems to codevelop from the start. Efforts to understand language without perception or perception without language attack with one boot off and substantially reduce the probability of developing a computational account of intelligence and subsequent theory-grounded applications.

Genesis/Cases: A substrate for reflective thinking about Macbeth

Simple plot summaries from Shakespeare provide anvils on which we hammer out our ideas. We use them in work on our Genesis/Cases system because they are familiar and because they are rich in universally important factors such as power, emotion, consequence, and ties between people. We have found that the same body of reflexive and reflective knowledge that works for Shakespeare works also for international conflict. Accordingly, some of our examples draw on the alleged 2007 Russian cyberattack on Estonia's network infrastructure.

Here, for example, is a brief rendering of *Macbeth*, part of a corpus under preparation by Capen Low:

Macbeth, Macduff, Lady Macbeth, and Duncan are persons. Macbeth is a thane and Macduff is a thane. Lady Macbeth, who is Macbeth's wife, is greedy. Duncan, who is Macduff's friend, is the king, and Macbeth is Duncan's successor. Macbeth defeated a rebel. Witches had visions and talked with Macbeth. The witches made predictions. Duncan became happy because Macbeth defeated the rebel. Duncan rewarded Macbeth because Duncan became happy. Lady Macbeth is greedy. Lady Macbeth, who is Macbeth's wife, wants to become the queen. Lady Macbeth persuades Macbeth to want to become the king. Macbeth murders Duncan. Lady Macbeth becomes crazy and dies. Dunsinane is a castle and Burnham Wood is a forest. Burnham Wood goes to Dunsinane. Macduff had unusual birth. Macduff fights with Macbeth and kills him. The predictions came true.

Easy inferences enable Genesis to put in what is left out, just as we humans make obvious inferences. For example, Duncan is dead because he was murdered, although his death is never stated. Such inferences arise from what we call background knowledge, also expressed in English. Here are a few representative examples, exactly as provided to Genesis:

James harmed Henry because James harmed George and George is Henry's friend. James wanted to become king because Henry persuaded James to want to become king. James may kill Henry because Henry is king and James is Henry's successor and James wants to become the king.

Equipped with these rule-like, reflexive statements, Genesis produces an *elaboration graph* of predictions and explanations, shown below, augmenting explicit elements provided in the summary, leading to more inferences (the gray boxes) than explicit elements. Note in the graph that Macduff's killing of Macbeth is explained as a consequence of Macduff disliking Macbeth. Fortunately, we do not always kill the people we dislike, but in the plot, as given, there was no other explanation, so the connection is supposed.

Given the elaboration graph, the system is ready to look for higher-level concepts of the sort we humans would see in the story but only if we reflect on what we read. Inspired by the pioneering work of Wendy Lehnert, we have arranged for our system to see a Pyrrhic victory in the elaboration graph for Macbeth: Macbeth wants to be king, murders Duncan to become king, but the murder leads to his own death:

For a more contemporary, suggestive example, Genesis finds a revenge in the elaboration graph produced from a description of the alleged Russian cyber attack on Estonia's network infrastructure.

Early on, we found the revenge pattern in the Estonia story using a page of complex code. In December, 2009, using a system built by David Nackoul, we were able to provide our system with an English description of revenge:

Start description of "revenge". XX and YY are entities. XX's harming YY led to YY's wanting to harm XX. YY's wanting to harm XX lend to YY's harming XX. The end.

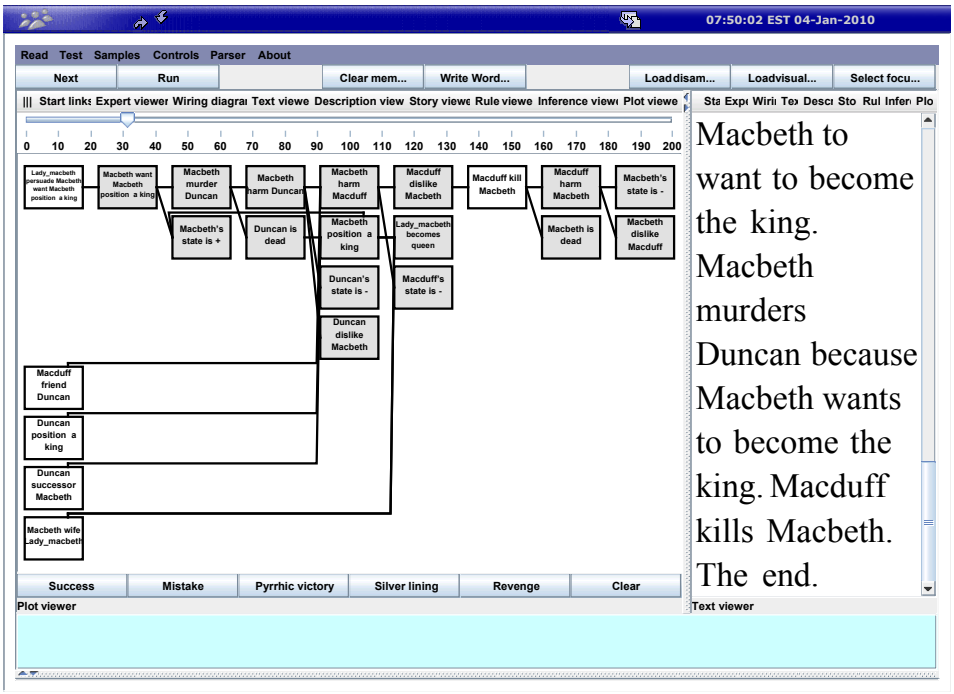

Figure 4. Genesis/stories produces an elaboration graph from English descriptions of reflexive knowledge together with a story. White boxes represent information given explicitly in the Macbeth story. Gray boxes represent information produced by reflexive knowledge via the connections shown between boxes.

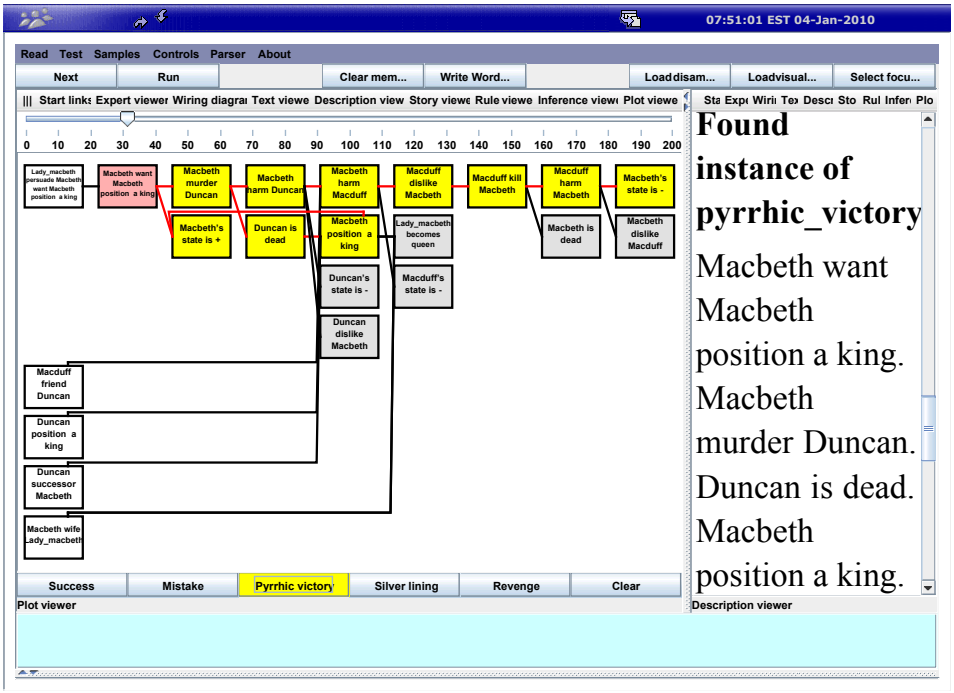

Figure 5. Genesis/stories uses the elaboration graph, together with reflective knowledge, to augment the explicit knowledge provided in the story and simple inferences enabled by reflexive knowledge.

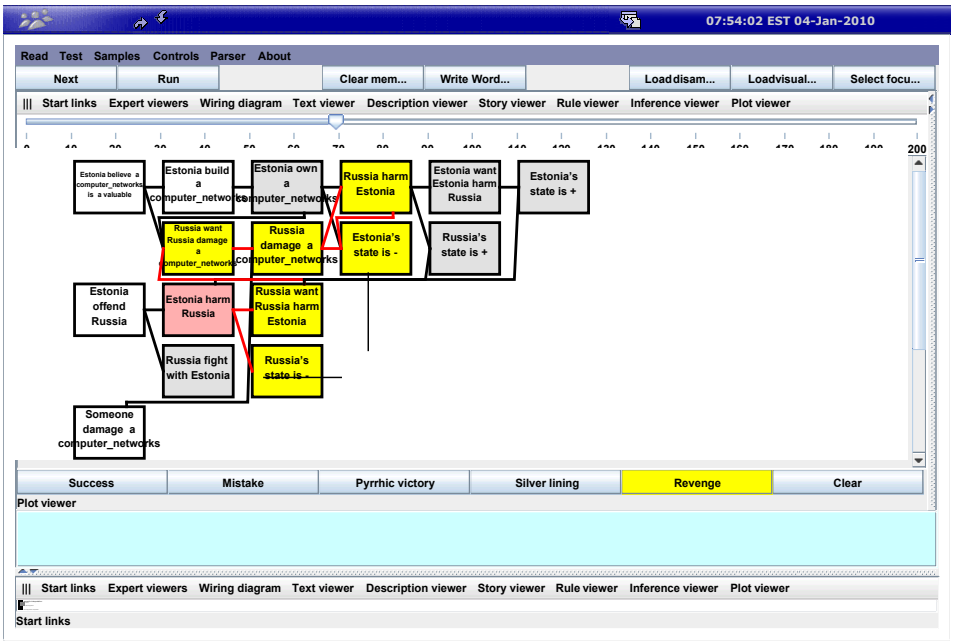

Figure 6. The reflexive and reflective knowlege honed on Macbeth have broad application. Here, the alledged Russian cyberattack on Estonia reveals an instance of *revenge*.

The English is not pretty yet, but the first lift-off of the capability is important. When this capability is fully developed, our Genesis/Cases system will deploy a large armamentarium of higher-level story descriptors.

In summary, we believe that if we are to develop a computational theory of human intelligence, we must understand how to build systems that use cases in every form, from fairy tales, through personal experience, and on to the great events in history. Our aim is to show that the issues are the same at every level, and that understanding the politics of the modern world, the intrigues of Shakespearean plots, and competitions among barnyard animals all depend on fundamentally the same computational machinery.

News

In the first half of 2010, we took Genesis to another level by arranging for the simultaneous reading of stories by two separate persona, jocularly call Dr. Jekyll and Mr. Hyde. Equipped with overlapping but slightly different points of view, Dr. Jekyll and Mr. Hyde see things differently.

In Figure 7, for example, Dr. Jekyll concludes that Macduff kills Macbeth in an act of insane violence; Mr. Hyde sees revenge. Both read the same story, but Dr. Jekyll thinks the only reason you would kill someone is that you are insane. Mr. Hyde looks for a reason, and sees anger.

Figure 8 shows another example in which Dr. Jekyll sees the Estonia matter as an act of revenge, because Dr. Jekyll considers Estonia a friend. Mr. Hyde, on the other hand, considers himself a friend of Russia, so the Estonian matter is a well-deserved teaching-a-lesson reaction.

The side-by-side rendering of story analysis has stimulated thinking about additional projects. Here are a few representative examples:

- How can Dr. Jekyll alter the story so as to bring Mr. Hyde's view more into congruence with Dr. Jekyll's view.

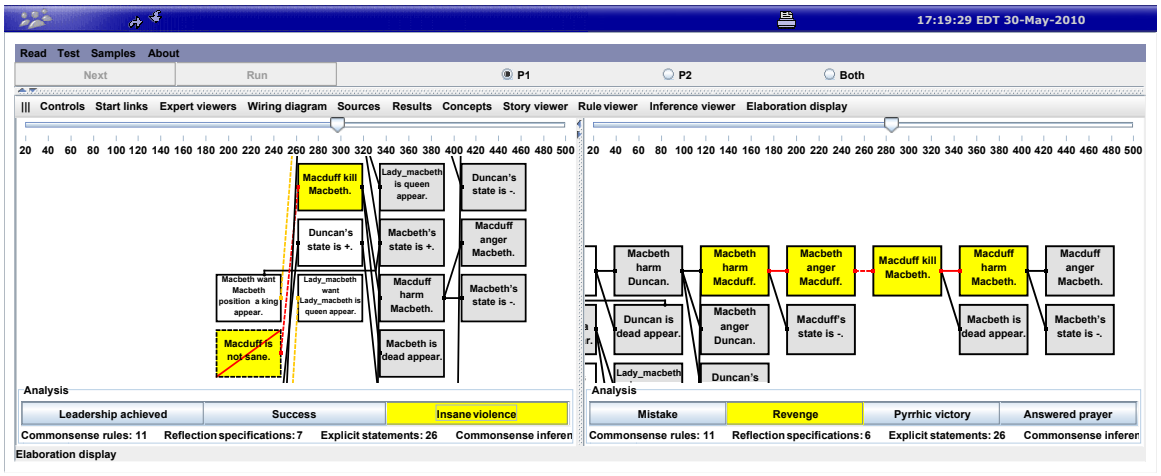

Figure 7. Opinions differ according to culture. One person's act of insane violence is another person's act of legitimate revenge.

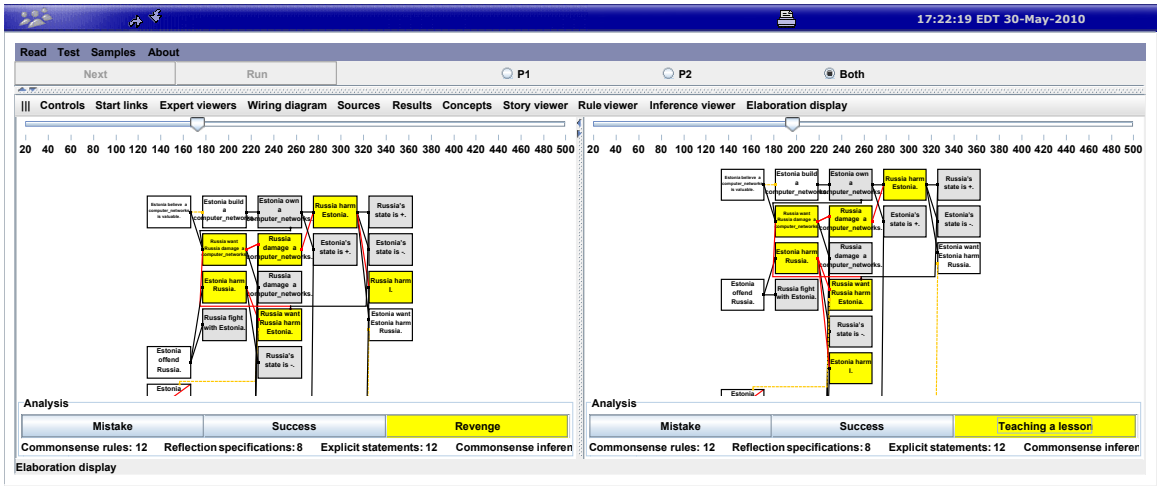

Figure 8. The proper label for the Russia-Estonia cyberattack depends on your allegiance. One person's revenge is another person's teaching a lesson.

- How can Dr. Jekyll tell Mr. Hyde what Mr. Hyde needs to know so as to do better analysis.
- How can an analyst determine if a response is an example of *tit for tat* or of *escalation* once a *revenge* or *teach-a-lesson* pattern has emerged.
- How can a negotiator help two sides to see things from the other point of view and bring them together.

Of course, once you can reflect on larger patterns, you can think about detecting their onset and intervening. Figure 9 illustrates by way of a snapshot taken by an onset detection system written by Adam Belay. Dr. Jekyll has noted, in the Estonia story, an action that can lead to revenge. In Belay's first-of-its-kind demonstration, the potential for a *mistake* is overreported because, as known to Genesis, any action may initiate a mistake. Thus, Belay's demonstration reveals automatic concept refining as yet another exciting research topic.

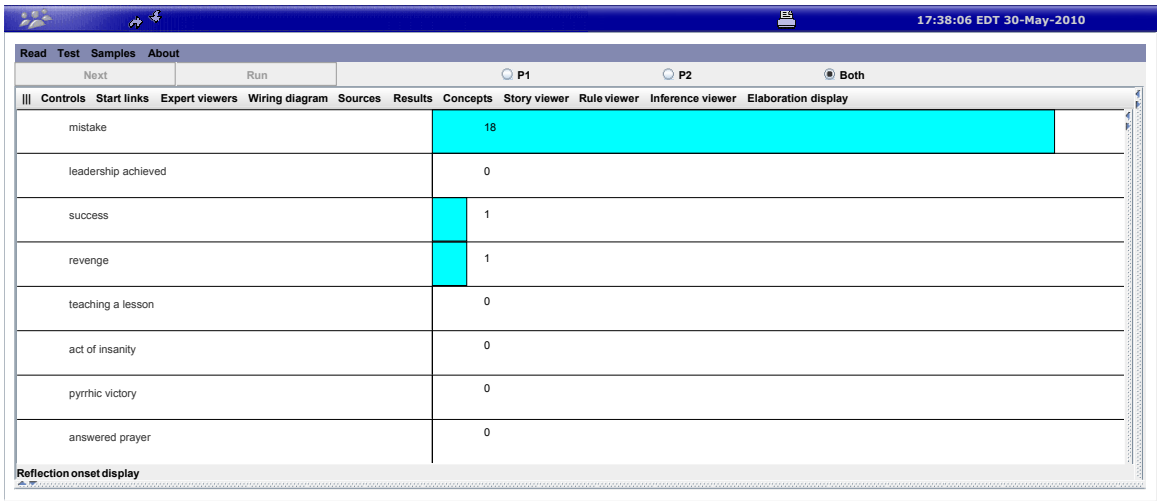

Figure 9. Detection of onset of dangerous situations. A first illustration of concept indicates the possibility of revenge.

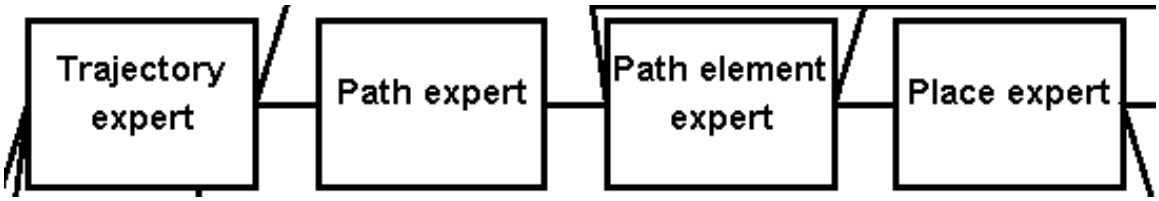

Figure 10. Genesis exploits ideas from Sussman's P3 architecture. The modules are boxes connected by wires. The architecture enables implementers to focus on individual boxes, eliminating the need to understand the system as a whole.

Pushing the Hardware and Software Substrate

In the 1960s and 1970s, early work in Artificial Intelligence provided the high challenges that attracted the people who put together personal computers, the Ethernet, the DARPANet, bitmap displays, and the forerunners of today's programming languages and programming-language environments. We can expect that history to repeat itself.

Already our Genesis system exploits some of the ideas in Gerald Susman's Propagator Programming Paradigm, P3. As prescribed by P3, the Genesis system consists conceptually of boxes connected by wires. Each box watches its ports and reacts to those signal-like objects it understands, transmitting new signals on the same or other ports. Each box ignores inputs it does not know how to handle.

Figure 10, for example, shows the system of trajectory, path, path-element, and place experts that handles a stream of parsed sentence fragments. Each expert takes from the stream what it recognizes and passes along what it does not. Simple statements put the wires in place using default ports:

```
Connections.wire( getTrajectoryExpert(),     getPathExpert());
Connections.wire( getPathExpert(),           getPathElementExpert());
Connections.wire( getPathElementExpert(),    getPlaceExpert());
```

P3 distantly resembles the abstraction and interface idea, but provides even more constraint and further reduces the programmer's radius of essential comprehension. Thus, the

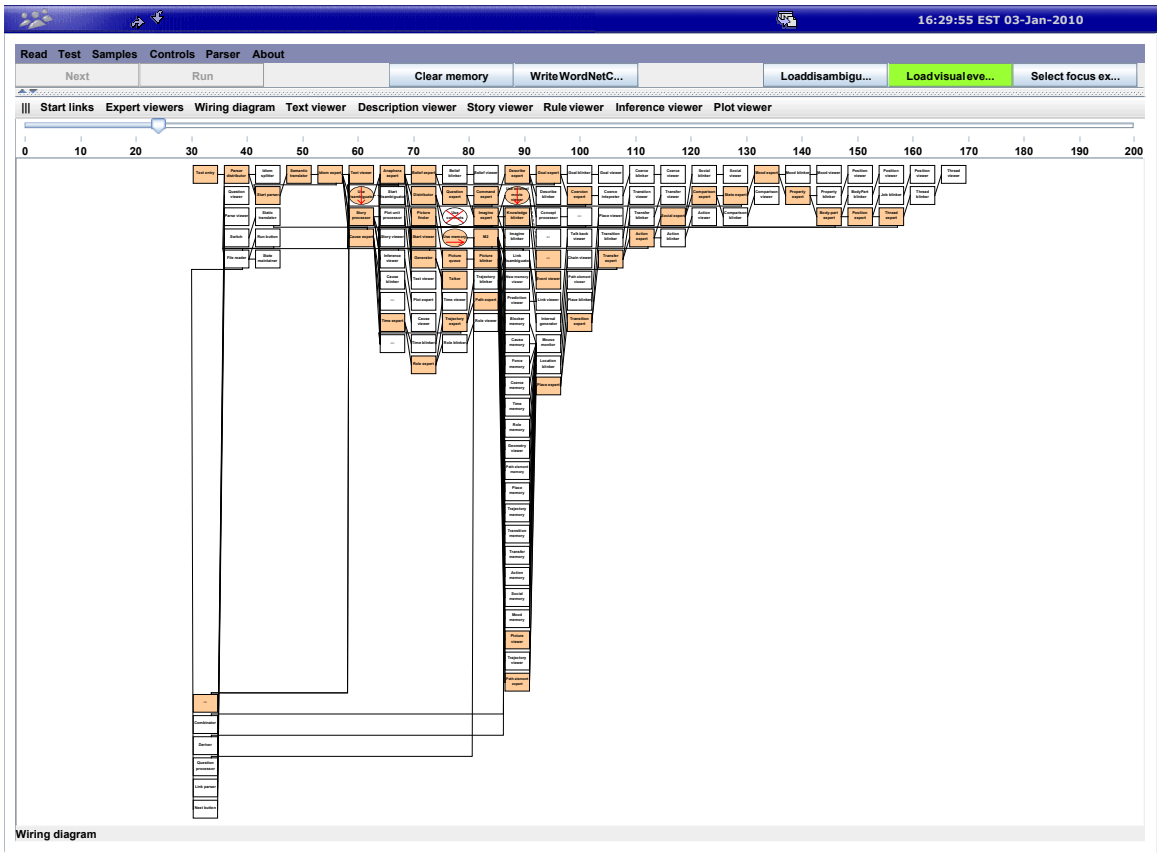

Figure 11. The Genesis system consists of more than 100 wired-together boxes, a number that increases with capability.

programmer in charge of a box need not understand all the details that lie beyond the box's ports, so the newly in-charge programmer can get started right away and never needs to understand the entire system shown in figure 11.

Because interaction is limited to propagation along wires, system architects can readily replace old boxes with better new boxes without the disruption that normally follows from module replacement. New boxes can run for a time in parallel with old boxes in a kind of shakedown mode.

In the future, we expect that S3 will further benefit by incorporating other ideas from P3, and in turn, will suggest further refinements to the paradigm. But this just scratches the surface.

This discussion of P3 is, of course, merely a for instance. A vigorous S3 program could motivate a great deal of leap-ahead thinking about hardware and programming. On the hardware side, for example, the Graph Machine championed by John Mallery is likely to prove indispensable, and on the software side, we hope for dramatic new languages and programming environments, rather than continued incremental evolution.

Contributions

We believe our work, and that of like-minded researchers, will have broad scientific impact comparable to the cracking of the genetic code. We have argued that the impact will emerge

from the following beliefs on language, perception, and connections between language and perception:

- Language enables guided perception
- Language stimulates perceptual imagination
- Guided perception and perceptual imagination are essential to thinking.

We have further argued that the scientific impact will emerge from the following beliefs on the role of language in enabling story telling, education, and cultural understanding:

- Language enables description, which enables story/case capture
- Much of education is surrogate story/case experience
- Surrogate experience determines culture.

So far, our particular contributions include the following:

- We have built a representation-centered language system.
- We have built a vision system that learns to recognize human activities such as jumping and giving.
- We have illustrated interaction between our language system and our answer-supplying vision system.
- We have shown how background knowledge can enrich explicit description in a story.
- We have demonstrated a system in which English descriptions of intermediate-level story concepts leads to higher-level story interpretation.

A program of larger scope would, of course, have immense practical impact, including the following:

- Economic advantage comparable to the development of the DARPANet via civilian analogs of military and intelligence systems
- New tools for winning the Global War on Terrorism through superior intelligence gathering and exploitation
- Improved conduct of international policy via systems capable of predicting unintended, culturally specific reactions to policies
- Cyber crime defense via intelligent network and machine monitoring
- Military advantage via smart robots, tactical, and strategic advisors.

Other benefits to programming and hardware innovation would be indirect but on the same scale of importance.